Application Report
SPRA723 - April 2001

Using the Flash Bootload Capability of the TMS320VC5402 DSK

Rob Gillis
Digital Signal Processing Solutions

ABSTRACT

Developing a routine for the TMS320VC5402 board is relatively straightforward. Creating a bootable routine and generating the Hex file to burn it into the Flash are also straightforward tasks. This short document walks a developer through the procedure in relatively simple terms.

Materials Needed

TMS320VC5402 DSP starter kit (DSK) and Code Composer Studio™ for the DSK, HEX500 utility (included with CCS) and the FLASHU utility (included with DSK).

Verify

Once a program has been developed, it should be verified functional by running it and checking to see if the results are correct.

Create a Vector Table

The vector table is used to identify the program's starting address after reset. Additionally, when the DSP is finished bootloading, it needs a known location to start from. The reset vector holds the address of where the program's code starts and will always be present. However, some development for the DSK can occur without a vector table (also known as an interrupt vector table). Generally, it is a good practice to have a vector table. Below is a basic vector table that can be used in a 'C' routine.

```
     .sect ”vectors”

     .global  _c_int00, start

start:    b     _c_int00
          nop
          nop
```

Verify the Memory Map and Location of Sections

For proper operation after bootloading, the memory map needs to be properly set up. Proper location of the vectors section, .data, .text, etc. is essential. The bootloader routine will place these sections in memory based on the information defined and linked. The default location for the interrupt vectors, for instance, is 0xFF80. However, depending on the configuration of the DSP and the system, this may not be a valid address. For the DSK on power up, MC/MP* is typically 0. This means that 0xFF80 is located in the on-chip ROM. This area cannot be written to. The interrupt vector table will most likely need to be relocated to internal memory at some RAM location. A good rule of thumb is to locate the vector table at 0x0080, then the program can start.

Verify the Operating Mode of the Memories Needed for the DSK

If the FLASHENB bit is set in the CNTL2 register (located at I/O location 0x4), then the external memory on the DSK is set to Flash. This enables the entire program and data sections to fit into the internal memory for proper operation. At bootup, the FLASHENB is set. This allows the bootload to occur from Flash.

For a non-BIOS program, the following Linker command file is a good example.

```
–c
–stack 0x0256
–heap 0x0256

MEMORY
{
   PAGE 0: VECS:     origin = 0080h, length = 0080h
           PROG:     origin = 200h, length = 2000h

   PAGE 1: DATA:     origin = 0x2200, length = 1e00h

}

/*******************************************************/
/* DSP Memory Allocation                               */
/*******************************************************/

SECTIONS
{
     .text     >    PROG PAGE = 0
     .data     >    DATA PAGE = 1
     .cinit    >    PROG PAGE = 0
     .bss      >    DATA PAGE = 1
      vectors  >    VECS PAGE = 0
}
```

The memory maps for the VC5402 DSK can be found in the Help files. Click on <Help><General Help>. Then under the contents tab, click on <TMS320C5402 DSK> <DSK memory maps and control registers>. There is information on the program memory map, the data and I/O memory maps.

Verify Project Build Options

It is important that the project options are set correctly to properly build your routine and make it bootable. One element that is critical in the setup of the project options is the –v 548 option. This is a compiler/assembler option that communicates to the hex utility which type of booloader to run. Any TMS320VC54x™ devices (including the VC5402) need this option set for proper creation of the boot tables and hex file.

Verify With These Modifications That the Routine Still Runs

It would be good practice at this point to recompile the routine and re-verify its functionality. Since the linker command file and the compiler/assembler options were changed, verifying the program's functionality at this point could help expose any mistakes that may have occurred.

Create a HEX File

Using the HEX500 utility create a bootable hex file for the routine. The following options are necessary for creating a bootable file.

```
yours.out /*  Name of obj file to be converted */
–m1          /*  Motorola s1 record format */
–memwidth 16  /*  Memory width target memory: 16 bits */
–romwidth 16  /*  Memory width Flash memory: 16 bits */
–boot        /*  create a boot table */
–bootorg 8000h /*  Boot info starts at 8000h in data */
–e 80h       /*  This is the entry point reset vector */
```

Note that the address for bootorg can start anywhere in the Flash. The entry point can be anywhere in the routine; however, it makes more sense to have it located at the reset vector. In the case above, the vector table has been relocated to 0x0080.

Creating a command file for the HEX500 utility is prudent to reduce typing mistakes. The above information is usable and valid for a HEXCOMMAND.CMD file.

Open an MS-DOS command window and go to the project directory. If a Hex Command file is created from the above, then execute the following:

```
C:\ti\MyProjects\yourproject>> HEX500 HEXCOMMAND.CMD
```

This will create a hex file for the project. The name will be YOURS.m00.

Program the Flash on the DSK

Finally, the routine is ready to be programmed into the DSK's Flash. From within the MS-DOS command window change directory to where the DSK Utilities are located. (typically: C:\ti\c5400\dsk\utilities)

Be sure to close Code Composer, power down the VC5402 DSK, Jumper Pins 1 and 2 on the JP2 connector and power up the VC5402 DSK. (Note: Pin 1 on JP2 is the pin on the end of the connector closest to the parallel port connector.)

Execute the following line:

```
FLASHU 2 –l c:\ti\myprojects\yourproject\yours.m00
```

Power down the VC5402 DSK. Remove the jumper on JP2. Re-power the VC5402 DSK and the routine should bootload and run.

Once the Flash on the DSK has completed, power down the VC5402 DSK. Remove the jumper on JP2. Re-power the VC5402 DSK and the routine will boot load and run.

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, license, warranty or endorsement thereof.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations and notices. Representation or reproduction of this information with alteration voids all warranties provided for an associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Resale of TI's products or services with *statements different from or beyond the parameters* stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Also see: Standard Terms and Conditions of Sale for Semiconductor Products. www.ti.com/sc/docs/stdterms.htm

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265